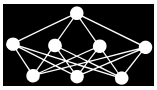

OPTIMAL LEARNING OF FUZZY NEURAL NETWORK USING ARTIFICIAL IMMUNE ALGORITHM

Dong Hwa Kim[*], *Ajith Abraham*[†]

Abstract: Fuzzy logic, neural network, fuzzy-neural networks play an important role in the linguistic modeling of intelligent control and decision making in complex systems. The Fuzzy-Neural Network (FNN) learning represents one of the most effective algorithms to build such linguistic models. This paper proposes an Artificial Immune Algorithm (AIA) based optimal learning fuzzy-neural network (IM-FNN). The proposed learning scheme includes the discovery of the fuzzy-neural network structure which can handle linguistic knowledge and the tuning of the membership function of the fuzzy inference system is achieved by AIA. The learning algorithm of the IM-FNN is composed of two phases. The first phase is to find the initial membership functions of the fuzzy neural network model. In the second phase, immune algorithm is used for tuning the membership functions of the proposed model. This paper also suggests techniques in determining the values of the steady-state equivalent circuit parameters of a three-phase squirrel-cage induction machine using immune algorithm.

Key words: *Fuzzy neural network, immune algorithm, induction motor, parameter estimation*

Received: November 26, 2007
Revised and accepted: May 5, 2008

1. Introduction

Some researchers suggest a model of fuzzy neuron that linear synaptic connections can be replaced with a nonlinearity characterized by a membership function and a fuzzy neural network model [1, 2]. The nonlinear characteristics are basically represented by fuzzy if-then rules with complementary membership functions.

[*]Dong Hwa Kim
Department of Instrumentation and Control Engineering, Hanbat National University, 16-1 San Duckmyong-Dong Yuseong-Gu, Daejon City, Korea, 305-719
[†]Ajith Abraham
Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, Trondheim, Norway, E-mail: ajith.abraham@ieee.org, http://www.softcomputing.net

Since fuzzy-neuron model or fuzzy-neural network possess good ability to describe a nonlinear relationship between multi-inputs and multi-output as well as its short leaning time compared with a conventional neural network, these models are expected as future linguistic tools for soft computing [11, 15, 16]. On the other hand, Radial Basis Function Networks (RBFN) and Backpropagation Neural Networks (BPNN) have yielded useful results in many practical areas such as pattern recognition, system identification and control, primarily due to their simple structures for realization and well established training algorithms [18]. Many fuzzy paradigms, meanwhile, have been studied is recent years by viewing a Fuzzy Logic System (FLS) as functionally equivalent to a RBFN or BPNN [17]. The most important advantage of such an FLS spanned by fuzzy basic functions is the provision of a natural framework for combining numerical values and linguistic symbols in a uniform way [3, 4]. From a mathematical point of view, the input-output expressions of those mappings are identical in spite of the distinct inference procedure. Capability discrimination between the neural and fuzzy system is thus diminished for proofs of universal neural/fuzzy approximators. Using neural networks or fuzzy systems to approximate a given plant or to control a process flow depends on whether rich available data are at hand or whether the *if-then* control heuristics could be established by human experts familiar with the system dynamics under consideration. A simple sigmoidal-like neuron is employed as a pre-assigned algorithm of the law of structural change which is directed by the current value of the error signal. However, in case of almost fuzzy logic systems and fuzzy-neural networks, the grade of the membership and the weighting function must be tuned by an approximation method or the experience-based tuning method. Some papers are written with a couple of objectives to demonstrate that the Genetic Algorithm (GA) is an efficient and the robust tool for generating fuzzy rules and the weighting function. The GA can construct a set of fuzzy rules that could optimize multiple criteria [5]. This paper proposes an artificial immune system [6, 7] based on optimal learning approach for designing fuzzy-neural [8-10, 14] network. The first phase of the IM-FNN is to find the initial membership functions of the fuzzy neural network model and the second phase is to obtain optimal membership functions of the proposed model by immune algorithm. This paper also deals with parameter estimation of an induction motor using the proposed method.

2. Structure of an Immune Algorithm Based Fuzzy-Neural Network

The structure of IM-FNN is illustrated in Fig. 1 [3] and the output of the FNN part of IM-FNN can be represented by (1). As illustrated in (1), the input space x_i is divided into several fuzzy segments which are characterized by membership functions μ_{i1}, $\mu_{i2}, ..., \mu_{in}$ within the range between $x_{\min}$ and $x_{\max}$. The grade of membership function is also given as numbers assigned to labels of a fuzzy membership function. The membership functions are followed by variable weights $w_{i1}, \ldots, w_{in}$. Mapping from x_i to $f_i(x_i)$ is determined by fuzzy inference and fuzzy rule as in (2).

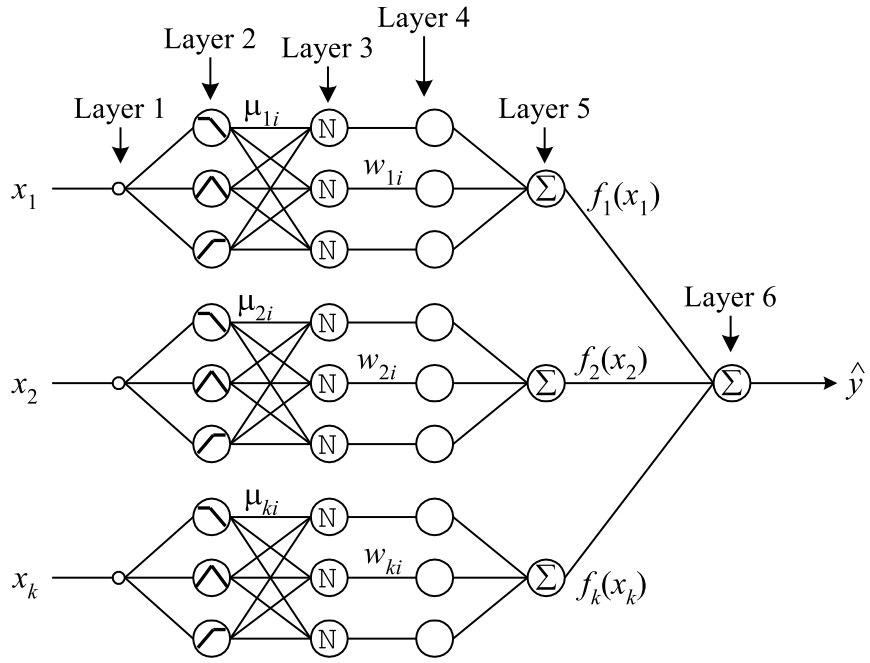

Fig. 1 *The structure of immune algorithm based optimal learning fuzzy-neural network.*

$$\begin{aligned} \overline{y} &= f_1(x_1) + f_2(x_2) + ... + f_m(x_m) \\ &= \sum_{i=1}^{m} f_i(x_i) \end{aligned} \tag{1}$$

$$\begin{aligned} R^1 &: \ If\ x_i\ is\ A_{i1}\ \ then\ \ C_{yx1} = w_{x1} \\ &\bullet \\ &\bullet \\ &\bullet \\ R^n &: \ If\ x_i\ is\ A_{in}\ \ then\ \ C_{yxn} = w_{xn} \end{aligned} \tag{2}$$

As the fuzzy inference adopted here is that of a singleton consequent, each weight w_{ij} is a deterministic value such as 0.8, 0.9. It should be emphasized that each membership function in antecedent is triangular and assigned to be complementary with neighboring ones. In other words, an input signal x_i activates only two membership functions simultaneously and the sum of grades of these two neighboring membership functions labeled by k and k+1 is always equal to 1, that is $\mu_{i,k}(x_i) + \mu_{i,k+1}(x_i) = 1$. So, the output of the fuzzy neural network can be

represented as follows:

$$\begin{aligned} f_i(x_i) &= \sum_{i=1}^{n} \mu_{xi} \bullet C_{yxi} \\ &= \frac{\sum_{j=1}^{n} \mu_{ij}(x_i) w_{ij}}{\sum_{j=1}^{n} \mu_{ij}(x_i)} = \frac{\mu_{ik}(x_i) w_{ik} + \mu_{ik+1}(x_i) w_{ik+1}}{\mu_{ik}(x_i) + \mu_{i,\,k+1}(x_i)} \\ &= \sum_{i=1}^{n} \mu_{xi} \cdot w_{xi} \end{aligned} \tag{3}$$

In (3), the weight w_{ij} is assigned by learning the rule which is described by n *if-then* rules. That is, If input x_i lies in the fuzzy segment μ_{ij}, then the corresponding weight w_{ij} should be increased directly proportional to the output error (y-$\overline{y}$), because the error is caused by the weight. This proposition can be represented as follows:

$$f_i(x_i) = \mu_{xi}(x_i) w_{xi} + \mu_{xi+1}(x_i) w_{xi+1} \tag{4}$$

The learning procedure is an incremental change of weights for each input pattern. That is, the incremental change of minimizing the squared error (4) is obtained from:

$$\Delta w_{xi}(t+1) = 2\delta \left(y - \overline{y}\right) \mu_{xi} + \alpha_i \left(w_{xi}(t) - w_{xi}(t-1)\right) \tag{5}$$

In this learning algorithm, all the initial weights are assigned to be zero and the updating of the weights is achieved after calculation of cumulative value in (5), where y is the given data, $\overline{y}$ is the output of model, δ learning rate, α is momentum constant and δ, α have the range of 0 to 1, respectively. w_{xi} is the present weighting function and $w_{xi}(t-1)$ is the previous weighting function.

3. Immune Algorithms for Obtaining Optimal Learning of the FNN

3.1 Immune algorithm

An artificial immune system is illustrated in Fig. 2. When an antibody on the surface of a B cell binds an antigen, B cell becomes stimulated. The level of stimulation depends not only on how well the B cell's antibody matches the antigen but also how it matches other B cells in the immune network. The stimulation level of the B cell also depends on its affinity with other B cells in the immune network. This network is formed by B cells possessing an affinity to other B cells in the system. If the stimulation level rises above a given threshold, the B cell becomes enlarged and if the stimulation level falls below a given threshold, the B cell dies off. The more neighbors a B cell has an affinity with, the more stimulation it will receive from the network and vice versa. Against the antigen, the level to which a B cell is stimulated relates partly to how well its antibody binds the antigen. We take into account both the strength of the match between the antibody and the antigen and the B cell object's affinity to the other B cells as well as its enmity.

Therefore, generally the concentration of i-th antibody, which is denoted by δ_i, is calculated as follows [6-8]:

$$\frac{dS_i(t)}{dt} = \left(\begin{array}{l} \sigma \sum_{j=1}^{N} m_{ji}\xi_j(t) \\ -\sigma \sum_{k=1}^{N} m_{ik}\xi_k(t) + \beta m_i - \gamma_i \end{array} \right) \delta_i(t) \tag{6a}$$

$$\frac{d\sigma_i(t)}{dt} = \frac{1}{1 + \exp\left(0.5 - \frac{dS_i(t)}{dt}\right)} \tag{6b}$$

In (6), N is the number of antibodies, and σ and β are positive constants. m_{ji} denotes affinities between antibody j and antibody i (i.e. the degree of interaction), m_i represents affinities between the detected antigens and antibody i, respectively.

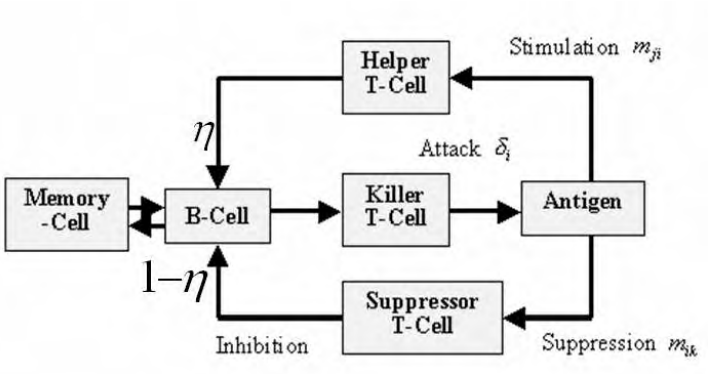

Fig. 2 *Dynamic relationship between cells and antigen in an immune system.*

3.2 Immune based membership function tuning

The initial value of the membership function (triangular) is given by X_1min= [0.46, 0.48], X_1max= [0.77, 0.81], X_2min = [45.0, 47.0] X_2max= [61.0, 63.0], and learning rate boundary δ = [0.001, 0.01] and momentum constant boundary α=[0.00001, 0.0004], respectively. The final membership function obtained by the immune algorithm is illustrated as a dashed line in Figs. 3(a) and (b).

3.3 Immune algorithm based on computational procedure for optimal selection of parameter of FNN structure

We used the immune algorithm based on calculation procedure as depicted in Fig. 4 to optimize the learning rate, momentum term and fuzzy membership function of IM-FNN. We used 10 and 100 generations, 60 population size, 10 bits per string, crossover rate equal to 0.6, and mutation probability equal to 0.1, respectively.

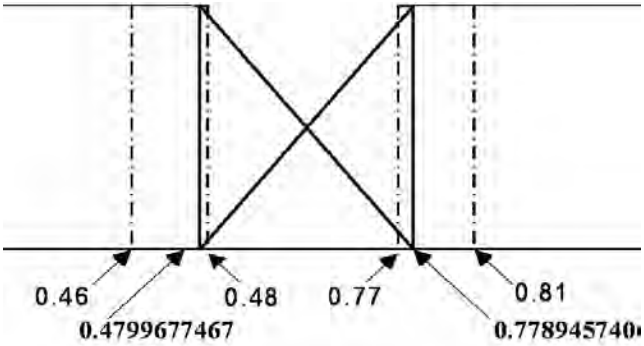

Fig. 3 *(a) Membership function shape of* x_1.

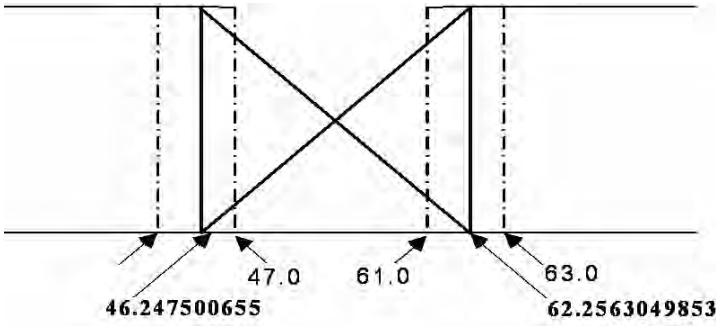

Fig. 3 *(b) Membership function shape of* x_1.

[Step 1] Initialization and recognition of antigen: The immune system recognizes the invasion of an antigen.

[Step 2] Product of antibody from the memory cell: The immune system produces the antibodies that were effective to kill the antigen in the past. This is implemented by recalling a past successful solution from memory cell. For each individual of the network population, calculate the fitness function using memory cell to membership function, learning rate and momentum constant.

[Step 3] Antibody with the best fitness value obtained by calculation for searching an optimal solution is stored in memory cell.

[Step 4] Differentiation of lymphocyte: The B-lymphocyte cell, the antibody that matched the antigen, is dispersed to the memory cells in order to respond to the next invasion quickly. This is achieved by selecting individuals using tournament selection and applying genetic operators (crossover and mutation) to the individuals of network.

[Step 5] Stimulation and suppression of antibody: The expected value η_k of the stimulation of the antibody is given by:

$$\eta_k = \frac{m_{\varphi k}}{\sigma_k} \quad , \tag{7}$$

where σ_k is the concentration of the antibodies.

An immune system can control the concentration and the variety of antibodies in the lymphocyte population. If the antibody obtains a higher affinity against an antigen, the antibody stimulates. However, an excessive higher concentration of an antibody is suppressed. Through this function, an immune system can maintain the diversity of search directions and obtain an optimal solution.

[Step 6] Calculate fitness value between antibody and antigen. This procedure can generate a diversity of antibodies by a genetic reproduction operator such as mutation or crossover. These genetic operators are expected to be more efficient than the generation of the antibodies.

[Step 7] If the maximum number of generations of the memory cell is reached, stop and return the fitness of the best individual fitness value to network; otherwise, go to Step 3.

4. Experiment Results and Discussions

4.1 Immune based optimal learning of FNN structure

In order to illustrate the learning effect of the proposed immune based FNN (IM-FNN), we use the second-order nonlinear difference equation given as [4]

$$y_k = \frac{y_{k-1}y_{k-2}(y_{k-1}+2.5)}{1+y_{k-1}^2+y_{k-2}^2} + u_k \tag{8}$$

Fig. 7 depicts the performance index by clonal differentiation rate of the immune algorithm for the given model (8) and Figs. 8 and 9 illustrate the best fitness and the object function depending on the differentiation rate of clonal selection pCS = 0.2 and clonal selection pCS = 0.5, respectively, when the number of Membership Function (MF) is 2. Fig. 10 shows comparison of the fitness value depending on the differentiation rate of clonal selection (pCS), when the number of the membership function is 2 (mem = [2, 2]). Fig. 11 represents the best value of the fitness function and object function, when the number of the membership function is 3 (mem = [3, 3]) and differentiation rate of clonal pCS = 0.2, respectively. Fig. 13 provided a comparison of the fitness value for pCS with respect to the number of the membership function, 3. Figs. 14–16 illustrate the best fitness value when learning parameter of the immune algorithm is 100 generations, 0.2, 0.5 pCS (differentiation rate of the clonal selection), and the number of the membership function is varied from 2 to 3. Figs. 17 and 18 illustrate the best fitness value when learning parameter

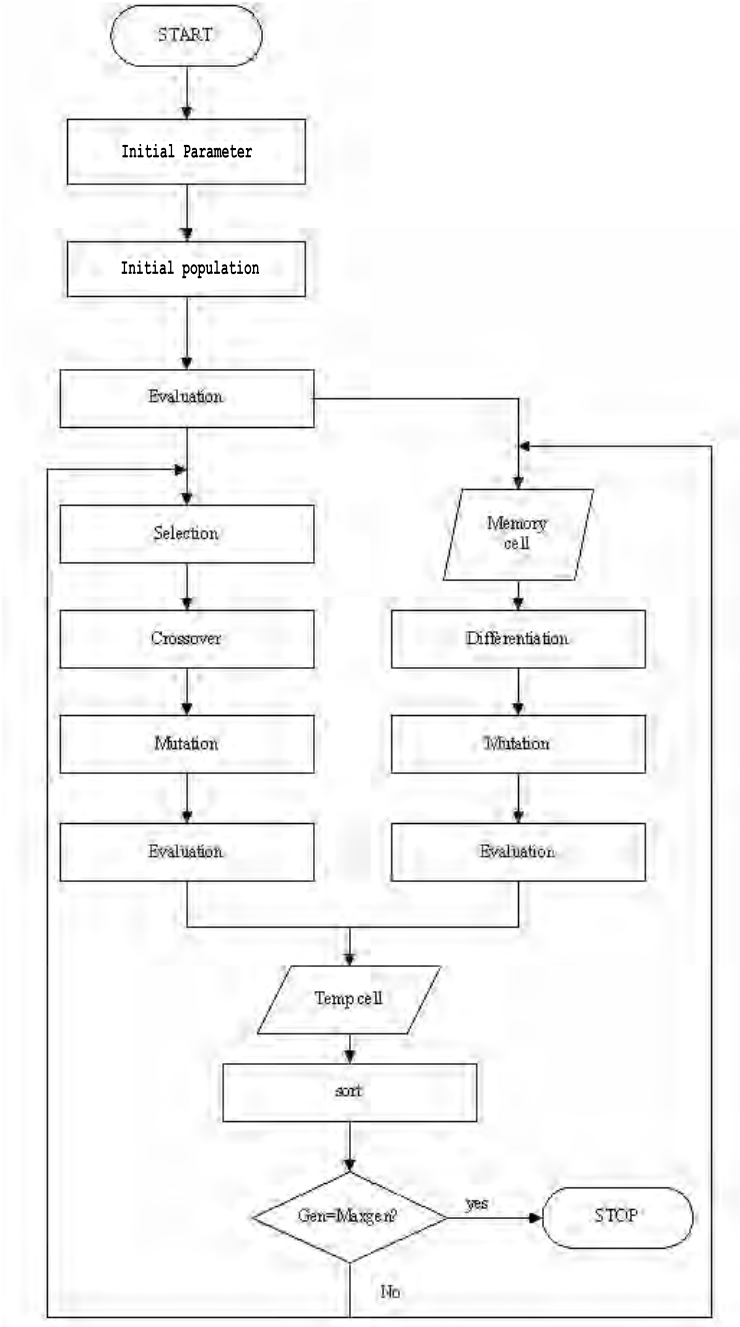

Fig. 4 *Computational procedure for parameter selection of FNN structure by clonal selection of immune algorithm.*

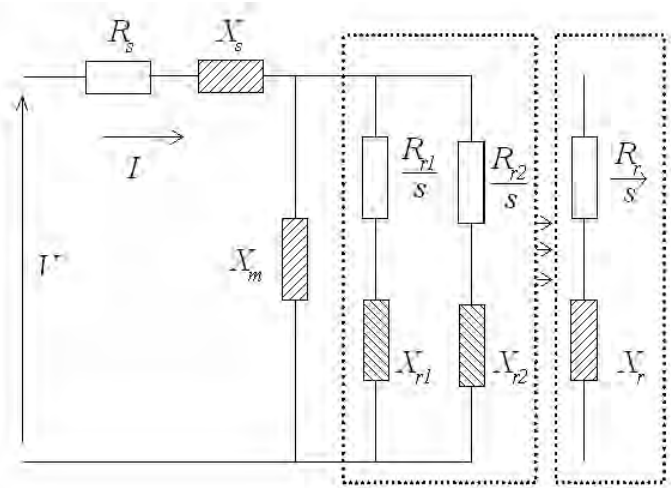

Fig. 5 *Equivalent circuit of a squirrel-double cage induction motor.*

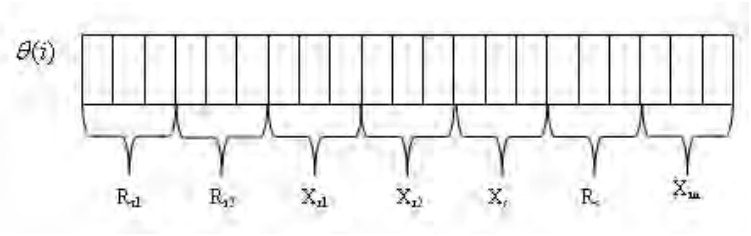

Fig. 6 *Allocation structure of each parameter in the immune system.*

pCS	2:2		3:2		3:3	
	PI	E_PI	PI	E_PI	PI	E_PI
0.2	0.0354	0.2857			0.0354	0.2857
0.3	0.0408	0.2729			0.0356	0.2855
0.4	0.0409	0.2726			0.0359	0.2852
0.5	0.0394	0.2742			0.0361	0.2847

Tab. I *Parameters obtained by simulation.*

of the immune algorithm is 100 generations and the number of the membership function is varied from 2 to 3 and for different values of pCS.

Figs. 19–22 illustrate the Performance Index Error (PI) and Test Index Error (E_PI), for different pCS values and for different member ship functions (2 and 3). Tab. I depicts the obtained values of PI and E_PI for different pCS values, and Tab. II shows the evolution of the membership function shape depending on generation of the immune algorithm. Tab. III depicts a performance compari-

son of the learning results obtained by the GA based FNN model, HCM (Hard C-Means) clustering algorithm and GA based FNN, and the immune based FNN model proposed in this paper. Tab. IV illustrates the empirical results for 10 and 100 generations of immune algorithm, respectively.

4.2 Optimal parameter estimation of induction motor by immune based FNN structure

1) Induction motor model

A squirrel-cage induction machine supplied with a three-phase symmetrical voltage source can be described using the equivalent circuit shown in Fig. 5. In case the stator current the input power, the equations of and the electromagnetic torque for a squirrel induction motor can be deduced from the circuit of Fig. 5 and are expressed as follows:

$$
\begin{gathered}
I(s) = V\sqrt{\frac{C^2+D^2}{A^2+B^2}} \\
P(s) = 3V^2\frac{AC-BD}{A^2+B^2} \\
T(s) = 3V^2\frac{p}{\omega}\frac{R_{fe}^2 R_r/s}{A^2} \\
A = R_s\left(1+\frac{X_r}{X_m}\right)+\left(1+\frac{X_s}{X_m}\right)\frac{R_s}{S} \\
B = X_r + X_s\left(1+\frac{X_r}{X_m}\right) - R_s\left(R_r/S\frac{R_r/S}{X_m}\right) \\
C = 1+\frac{X_r}{X_m} \\
D = \frac{R_r/S}{X_m} \\
R_r = \frac{R_{r1}R_{r2}(R_{r1}+R_{r2})+(R_{r1}X_{r2}^2+R_{r2}X_{r1}^2)S^2}{(R_{r1}+R_{r2})^2+(X_{r1}+X_{r2})^2S^2} \\
R_r = \frac{X_{r1}X_{r2}(X_{r1}+X_{r2})S^2+R_{r1}^2X_{r2}+R_{r2}^2X_{r1}}{(R_{r1}+R_{r2})^2+(X_{r1}+X_{r2})^2S^2} \\
I(s) = V\sqrt{\frac{C^2+D^2}{A^2+B^2}} \\
P(s) = 3V^2\frac{AC-BD}{A^2+B^2} \\
\theta = [R_{r1}\ R_{r2}\ X_{r1}\ X_{r2}\ X_s\ R_s\ X_m]^T
\end{gathered}
\tag{9}
$$

In the above equations, R_s, R_r, and R_{fe} are the stator, rotor, and iron losses, respectively. Also, X_s, X_r, *and* X_m are the stator leakage reactance, rotor leakage reactance, and magnetizing reactance. For neglecting the iron losses of a double-cage motor, one must add a second branch in parallel with the magnetizing reactance.

2) Optimal parameter selection for motor parameters

In order to determine model parameters from the slip curves of the equivalent circuit [13], Lima et al. [12] used a nonlinear curve-fitting problem stated as the solution of the following minimization problem

$$\min_{\theta\in\Omega} = J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[y_i - y\left(s_i,\ \theta\right)\right]^2 \tag{10}$$

where $J(\theta)$ is the least squares cost function obtained by the sum of the squares of the differences between the experimental and calculated slip curves, θ is parameter space depending on the number of parameters to be estimated, y_i is the experimental data value collected from machine, $y(s_i,\ \theta)^2$ is nonlinear function relating the measured data, the circuit parameters, and the slip, and θ is parameter vector pertaining variation of the slip. Therefore, in case of double cage, dimension of parameter vector θ is defined as:

$$\theta = \left[R_{r1},\ R_{r2},\ X_{r1},\ X_{r2},\ X_s,\ R_s,\ X_m\right]^T. \tag{11}$$

Equation (11) depends on the kind of available experimental data and for obtaining a parameter vector that minimizes the quadratic performance index defined by equation (10). In this case, since one must deal with a nonlinear algorithm to acquire the desired solution, some numerical problems may arise or a direct approach would require writing down the normal equations for solving them. The methods for numerical minimization of performance index (10) might be modified to update the estimated parameter vector according to a load change. The clonal selection algorithm suggested in this paper is simulated and compared with genetic algorithm, recursive algorithm by Lima et al. [12]. Lima et al. [12] used objective function $J_1(\theta)$ and object function $J_2(\theta)$ is introduced for more optimal parameter selection as follows:

$$J_1(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[I(si) - I(si,\theta)\right]^2 + \frac{1}{N}\sum_{i=1}^{N}\left[P(si) - P(si,\theta)\right]^2 \tag{12}$$

$$J_2(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[S(i)I(si) - S(i)I(si,\theta)\right]^2 + \frac{1}{N}\sum_{i=1}^{N}\left[S(i)P(si) - S(i)P(si,\theta)\right]^2 \tag{13}$$

Fig. 6 represents the allocation structure of each parameter in the immune system to obtain optimal parameter estimation. Fig. 23 illustrates the performance of Clonal selection when compared with the true values. Fig. 24 illustrates the power curve performance of Clonal selection when compared with the true values and Fig. 25 depicts the variation of P(s) obtained by Clonal selection (CS-GA), GA, recursive, and true values. Figs. 26–29 also provide various illustrations showing the performance of the proposed algorithm. Figs. 27–29 depict that objective function $J_2(\theta)$ obtained better satisfactory results than objective function $J_1(\theta)$

using the proposed CS-GA method. As evident from all the illustrations, the proposed algorithm is efficient and can be used in parameter estimation. Empirical results obtained are illustrated in Tabs. V-VIII.

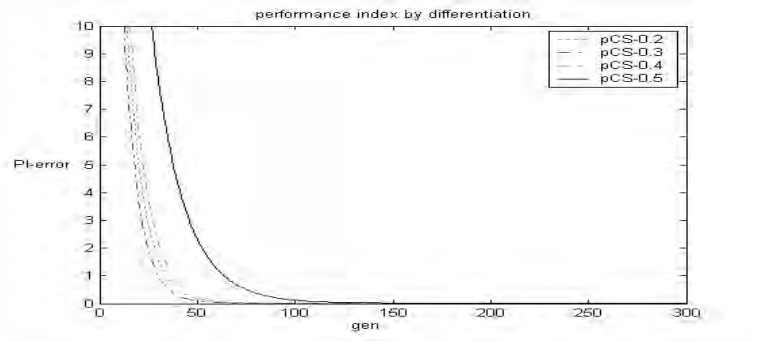

Fig. 7 *Performance index by differentiation rate of immune algorithm.*

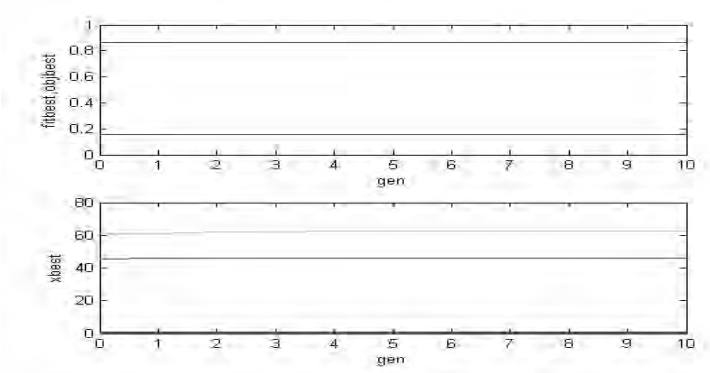

Fig. 8 *Best fitness value and object function (MF = [2, 2], pCS = 0.2).*

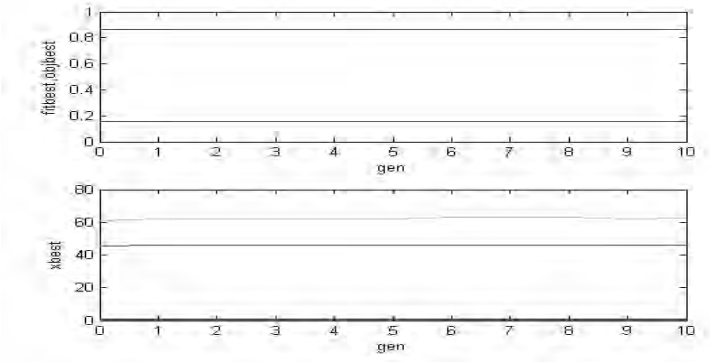

Fig. 9 *Best fitness value and object function (MF = [2, 2], pCS = 0.5).*

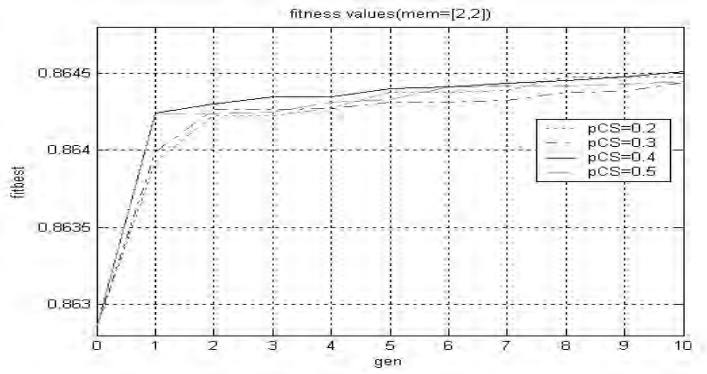

Fig. 10 *Comparison of fitness value depending on pCS (MF = [2, 2]).*

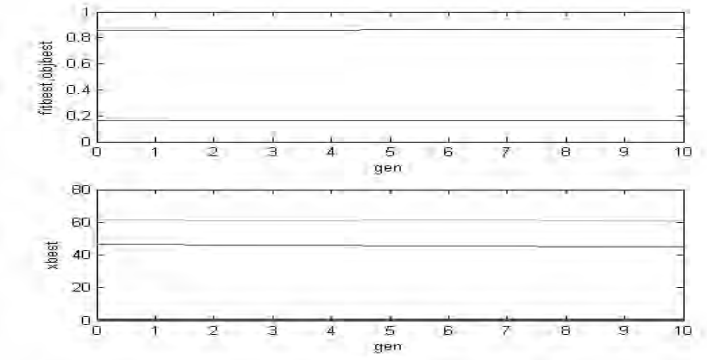

Fig. 11 *Best fitness value and object function (MF = [3, 3], pCS = 0.2).*

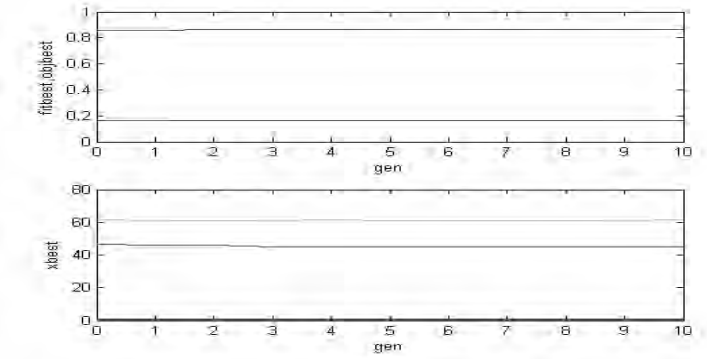

Fig. 12 *Best fitness value and object function (MF = [3, 3], pCS = 0.5).*

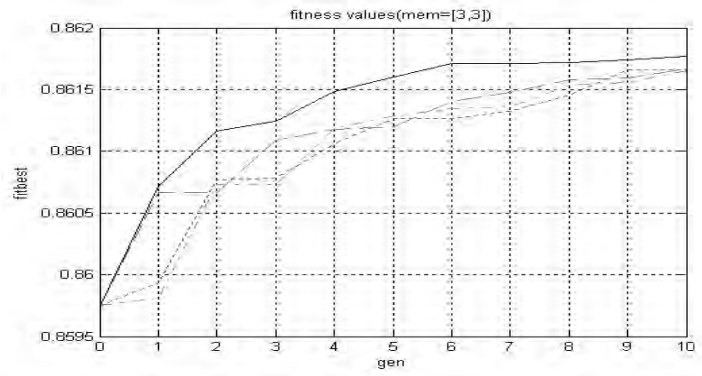

Fig. 13 *Comparison of fitness value for different values of pCS (MF = [3, 3]).*

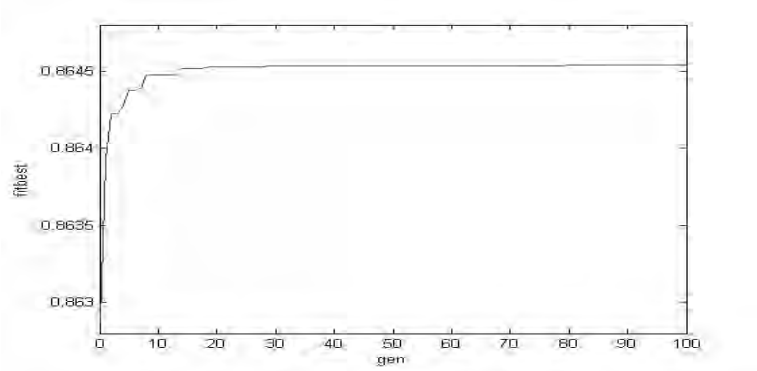

Fig. 14 *Best value of fitness (generations = 100, pCS = 0.2, MF = [2, 2]).*

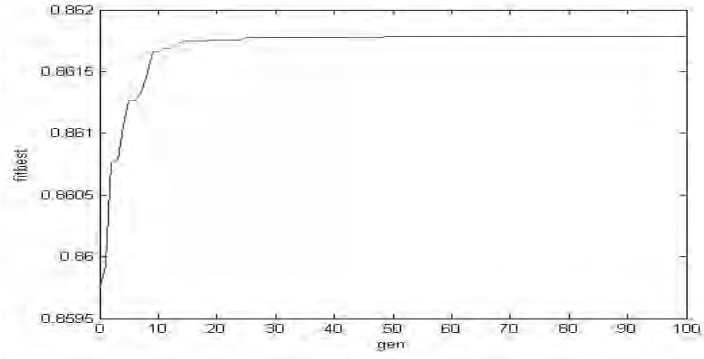

Fig. 15 *Best value of fitness (generations = 100, pCS = 0.2, MF = [3, 3]).*

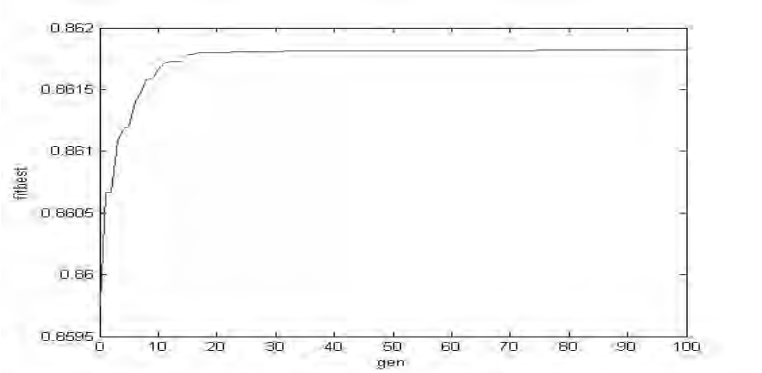

Fig. 16 *Best value of fitness (generations = 100, pCS = 0.5, MF = [3, 3]).*

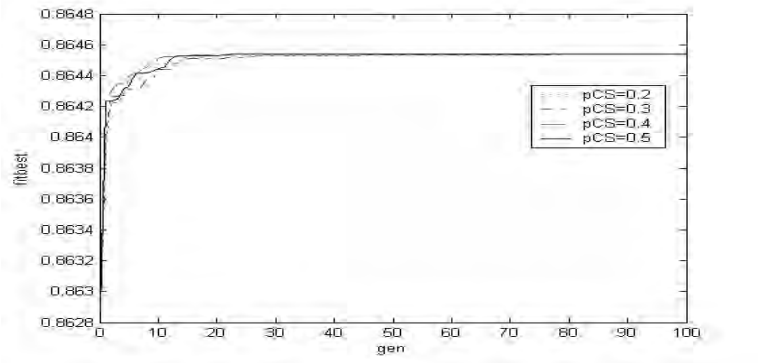

Fig. 17 *Best value of fitness for different pCS values (MF = [2, 2]).*

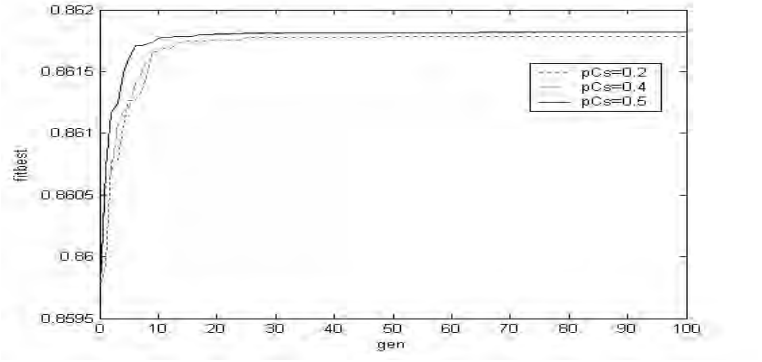

Fig. 18 *Best value of fitness by pCS. (MF = [3, 3]).*

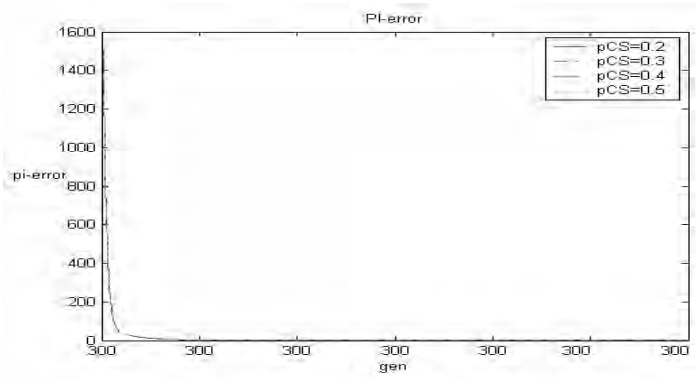

Fig. 19 *(a) PI-error for different values of pCS (MF = [2, 2]).*

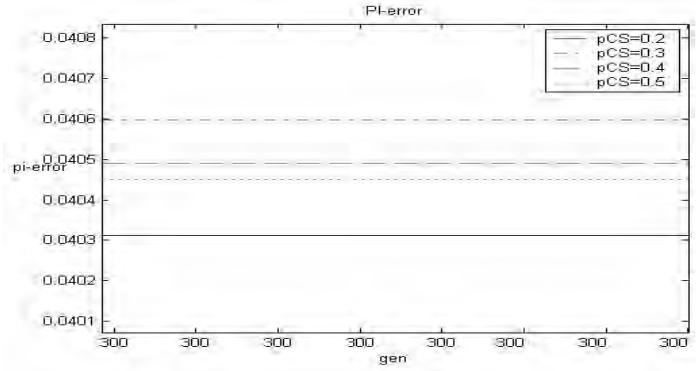

Fig. 19 *(b) PI-error for different values of pCS (MF = [2, 2]).*

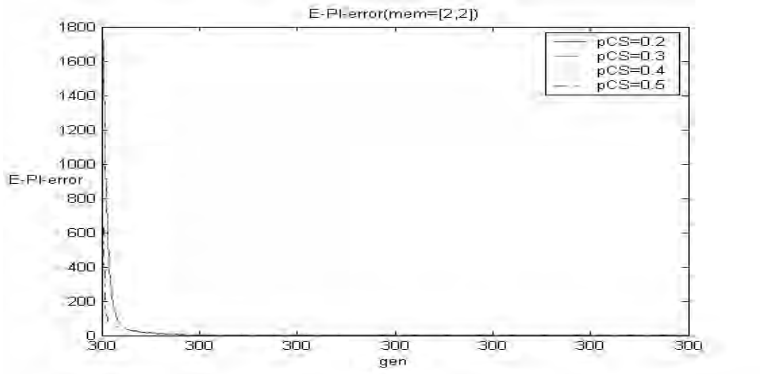

Fig. 20 *(a) E_PI-error for different values of pCS (MF = [2, 2]).*

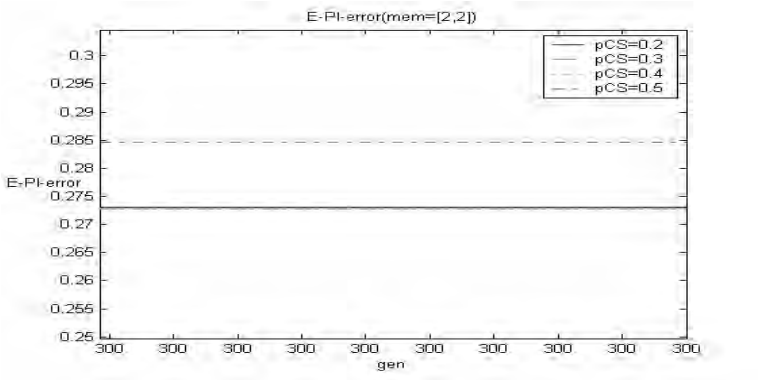

Fig. 20 *(b) E_PI-error for different values of pCS (MF = [2, 2]).*

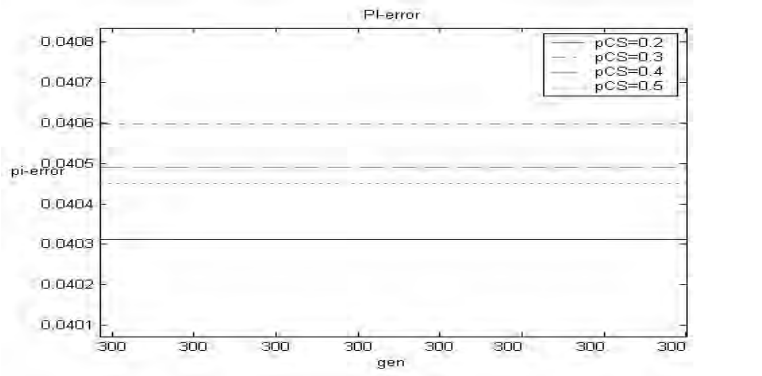

Fig. 21 *PI-error for different values of pCS (MF = [3, 3]).*

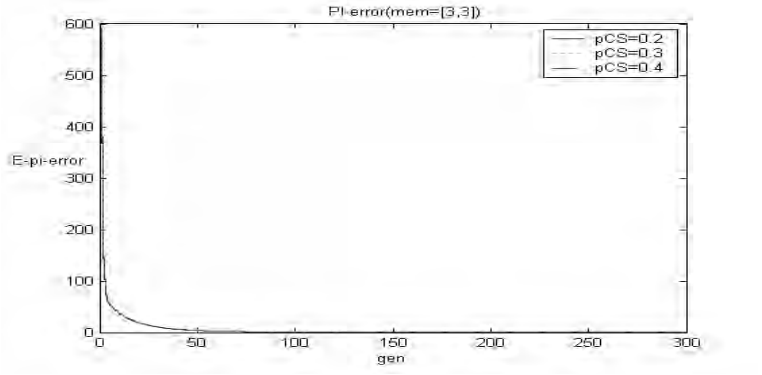

Fig. 22 *(a) E_PI-error for different values of pCS (MF = [3, 3]).*

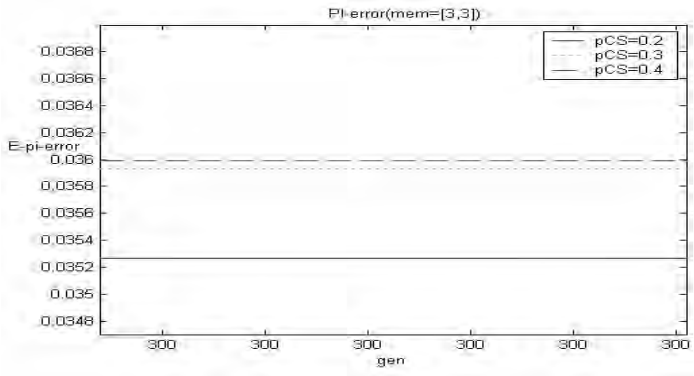

Fig. 22 *(b) E_PI-error for different values of pCS (MF = [3, 3]).*

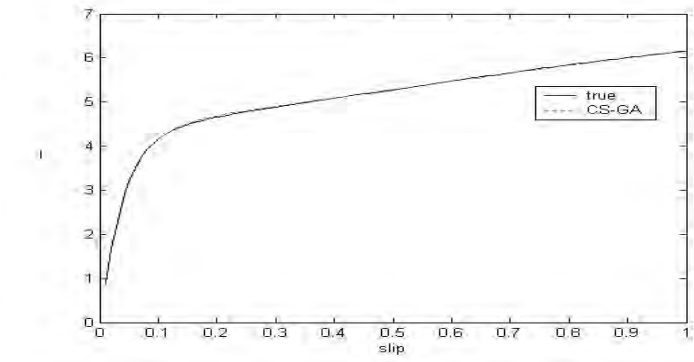

Fig. 23 *Variation of I(s) by Clonal selection and true values.*

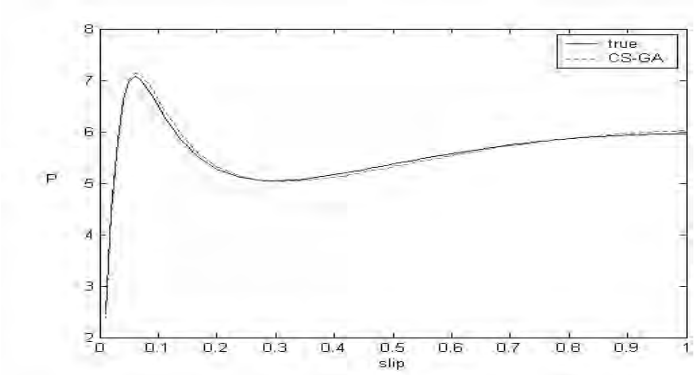

Fig. 24 *Power curve to slip by Clonal selection and true values.*

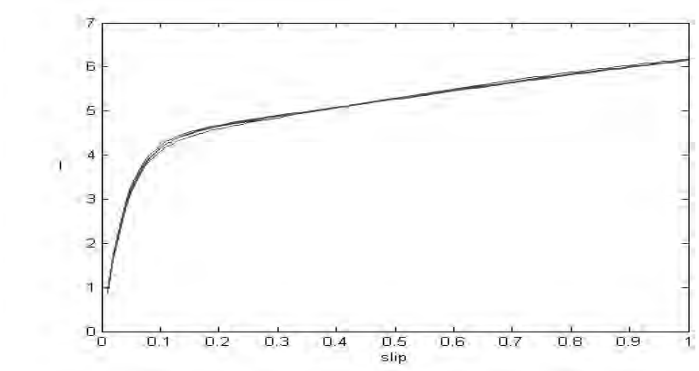

Fig. 25 *Variation of I(s) by Clonal selection, GA, recursive, and true values.*

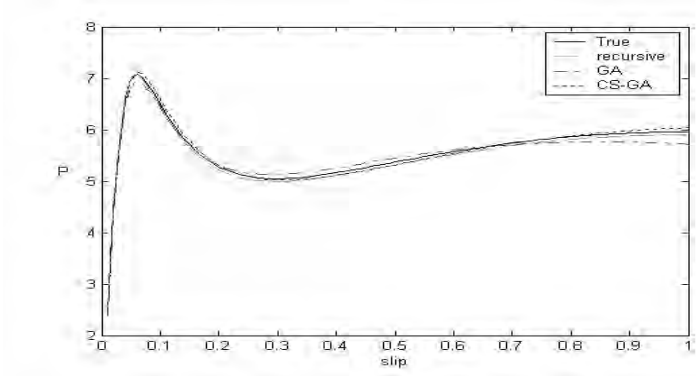

Fig. 26 *Variation of P(s) by Clonal selection, GA, recursive, and true values.*

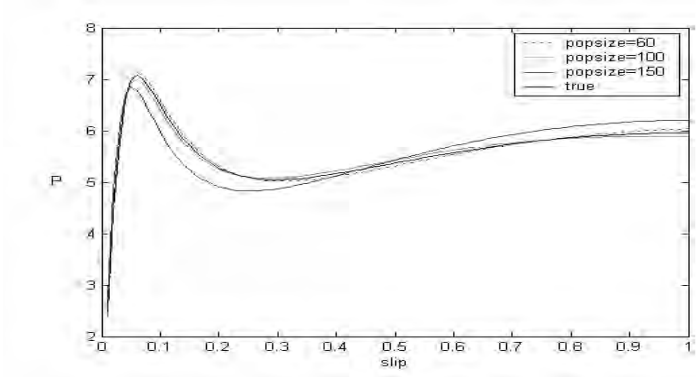

Fig. 27 *Parameter variation to population size for objective function* $J_1(\theta)$.

Item	X_1_min	X_1_max	X_2_min	X_2_max	δ	α	PI	E_PI
Ai	[0.46, 0.48]	[0.77, 0.81]	[45.0, 47.0]	[61.0, 63.0]	[0.001, 0.001]	[0.00001, 0.0004]		
A_0	0.4799677467	0.7789457406	46.247500655	62.2563049853	0.00100149345	0.0000354499	0.040311	0.27306
A_1	0.4799938583	46.2618954295	0.7814936556	62.4140905514	0.00100555325	0.0000465945	0.040598	0.27277
A_2	0.4799995994	46.2495033736	0.7825169491	62.5021071454	0.00100012016	0.0000197420	0.040491	0.27288
A_3	0.4799996948	46.2499020098	0.7813410294	62.4214891638	0.00100003433	0.0000420978	0.040452	0.27292
A_4	0.4600006103	45.0051097918	0.7700020980	61.0629330281	0.00268751019	0.0000279584	0.03265	0.28551
A_5	0.4600009918	45.0002574923	0.7700003814	61.2041646997	0.00262390100	0.0002727694	0.035923	0.28475
A_6	0.4600003242	45.0000438690	0.7700003814	61.2088834847	0.00261991846	0.00039791196	0.035983	0.28469

A_0gen=[100], mem=[2, 2], pCS=0.2, A_1gen=[100], mem=[2, 2], pCS=0.3, A_2gen=[100], mem=[2, 2], pCS=0.4
A_3gen=[100], mem=[2, 2], pCS=0.5, A_4gen=[100], mem=[3, 3], pCS=0.2, A_5gen=[100], mem=[3, 3], pCS=0.4
A_6gen=[100], mem=[3, 3], pCS=0.5

Tab. II *Membership function shape depending on generation of immune algorithm, the number of membership function and value of pCS.*

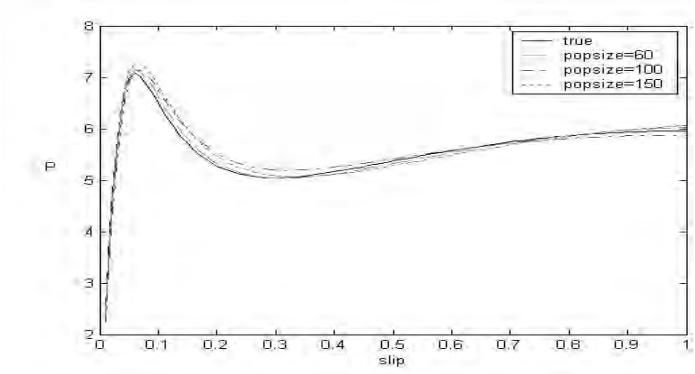

Fig. 28 *Response to variation of population size for object function* $J_2(\theta)$.

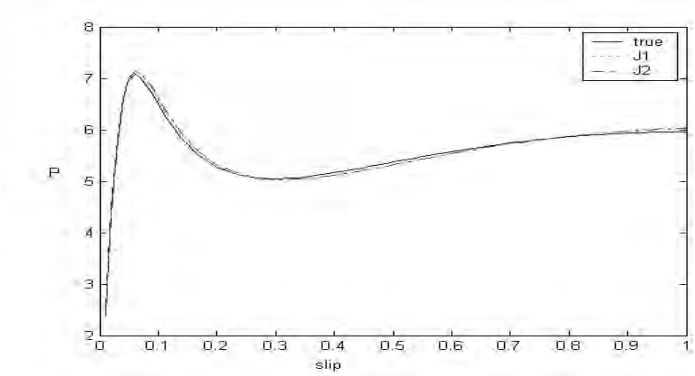

Fig. 29 *Comparison of two object function* $J_1(\theta), J_2(\theta)$.

Model	PI	E_PI	MF
FNN model (GA)	0.027	0.298	4
	0.026	0.304	6
FNN model (HCM+GA)	0.027	0.294	4
	0.032	0.276	6
FNN model (CS-GA)	0.0394	0.274	4
	0.0361	0.284	6

Tab. III *Comparison of the results by learning methods.*

5. Conclusions

Since Fuzzy sets and fuzzy logic can capture the approximate and qualitative aspects of human reasoning and decision-making processes, they have been considered

as effective tools to deal with uncertainties in terms of vagueness, ignorance, and imprecision. On the other hand, neural networks (NN) appeared as promising tools (or designing high performance control systems), because they have the potential for dealing with favorable scenarios owing to nonlinear dynamics, drift in plant parameters, and shifts in operating points. Since then, the fuzzy-neural network (FNN) learning represents one of the most effective algorithms to build such linguistic models for control system or making decision. However, in many case, tuning of membership and weighting function often remain as a difficult challenge.

This paper proposed an optimal learning method of fuzzy-neural network by the artificial immune algorithm. The learning algorithm based on an artificial immune system is used to design FNN (IM-FNN) and is capable of finding the initial membership functions and tuning of membership functions. The empirical results obtained are compared with the results with GA (genetic algorithm based neural network) and fuzzy-neural network, respectively. Empirical results clearly illustrate that the proposed learning method obtained more satisfactory results than other learning schemes.

This paper also introduced the proposed learning structure for an optimal parameter estimation of the induction motor and the results are compared with the conventional estimation method such as GA, recursive method.

	Gen=10				Gen=100			
pCS	2:2		3:3		2:2		3:3	
	PI	E_PI	PI	E_PI	PI	E_PI	PI	E_PI
0.2	0.0354	0.2857	0.0354	0.2857	0.040311	0.27306	0.035265	0.28551
0.3	0.0408	0.2729	0.0356	0.2855	0.040598	0.27277		
0.4	0.0409	0.2726	0.0359	0.2852	0.040491	0.27288	0.035923	0.28475
0.5	0.0394	0.2742	0.0361	0.2847	0.040452	0.27292	0.035983	0.28469

Tab. IV *Comparison of learning methods for 10 and 100 generations.*

gen	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
θ_{ture}	0.0693	0.0132	0.00843	0.1162	0.123	0.00778	4.3
xlb	0.06	0.01	0.007	0.10	0.10	0.006	4
xub	0.08	0.015	0.01	0.13	0.13	0.008	4.5

Tab. V *Initial boundary and true values.*

	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
Recursive	0.078	0.0129	0.0164	0.121	0.1167	0.0073	4.29
GA	0.063	0.0138	0.010	0.122	0.1260	0.0074	4.21
CS-GA	0.0755	0.0135	0.009	0.1168	0.1195	0.0064	4.34

Tab. VI *Comparison of parameter values obtained by each estimation methods.*

Obj_func	popsize	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
$J_1(\theta)$	60	0.0755	0.0135	0.0089	0.1168	0.1195	0.0064	4.34
	100	0.0647	0.0132	0.0081	0.1186	0.1253	0.0078	4.32
	150	0.0699	0.0115	0.0074	0.1231	0.1207	0.0066	4.42
$J_2(\theta)$	60	0.0755	0.0135	0.0091	0.1159	0.1196	0.0068	4.28
	100	0.068	0.0149	0.0081	0.1174	0.1242	0.0069	4.47
	150	0.0787	0.0136	0.0082	0.1105	0.1188	0.0077	4.30

Tab. VII *Resulting parameter estimation for each objective function and population sizes.*

Obj_func	gen	R_{r1}	R_{r2}	X_{r1}	X_{r2}	X_s	R_s	X_m
$J_1(\theta)$	100	0.075594	0.013538	0.0089178	0.11685	0.11951	0.0064439	4.3419
	150	0.072702	0.01227	0.007611	0.11916	0.1202	0.0069193	4.1168
	200	0.072702	0.01227	0.007611	0.11916	0.1202	0.0069193	4.1168
	300	0.072702	0.01227	0.007611	0.11916	0.1202	0.0069193	4.1168
$J_2(\theta)$	100	0.075521	0.013558	0.0091887	0.11596	0.11962	0.0068187	4.2845
	150	0.075521	0.013558	0.0091887	0.11596	0.11962	0.0068187	4.2845
	200	0.075495	0.013516	0.0092333	0.11596	0.11961	0.0068168	4.4295
	300	0.0755	0.01356	0.0092373	0.11596	0.11963	0.0068197	4.3595
	400	0.0755	0.01356	0.0092379	0.11596	0.11963	0.0068197	4.3604

Tab. VIII *Resulting parameter values for each object functions and different generations.*

References

[1] Park S., Sandberg I. W.: Approximation and radial-basis-function networks, Neural Computer, pp. 105-110.

[2] Lee C. C.: Fuzzy logic in control system: Fuzzy logic controller, part I and II, IEEE Trans. Syst. Man Cybern, **20**, 2, 1990, pp. 404-435.

[3] Wang H., Brown M., Harris C. J.: Neural network modeling of unknown nonlinear systems subiect to immeasurable disturbances, IEE Proc., Control Theory Appl., **141**, 4, 1994, pp. 216-222.

[4] Horikawa S. Furuhashi T., Uchikawa Y.: On fuzzy modeling using fuzzy neural networks with back propagation algorithm, IEEE Trans. Neural network, 1992, pp. 801-806.

[5] Farag W. A., Quintana V. H., Germano Lambert-Torred: A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems, IEEE Trans. on neural networks, **9**, 5, Sept. 1998.

[6] Farmer J. D., Packard N. H., Perelson A. S.: The immune system, adaptation, and machine learning, Physica, **D**, 22, 1986, pp. 187–204.

[7] Mori K., Tsukiyama M.: Immune algorithm with searching diversity and its application to resource allocation problem, Trans. JIEE, **113-C**, 10, 1993.

[8] Kim D. H., Hong W. P., Park J. I.: Auto-tuning of reference model based PID controller using immune algorithm, IEEE international conference on evolutionary computation, Hawaii, 2002, pp. 483-488.

[9] Kim D. H.: Intelligent tuning of a PID controller using an immune algorithm, Transactions of KIEE, **51-D**, 1, 2002, pp. 78-91.

[10] Kim D. H.: PID Controller Tuning of a Boiler Control System Using Immune Algorithm Typed Neural Network, 4th International Conference Computational Science, Lecture Notes in Computer Science, 2004, pp. 695-698.

[11] Park B. J.: Fuzzy polynomial neural networks: Hybrid architectures of fuzzy modeling, IEEE Trans. on Fuzzy systems, **10**, 5, 2002, pp. 607-621.

[12] Lima A. M. N., Jacobina C. B., Souza Filho E. B.: Nonlinear parameter estimation of steady-state induction machine models, IEEE Trans. Industrial Electronics, **44**, 3, 1997.

[13] Hickiewicz J., Macek-Kaminska K., Wach P.: Algorithmic methods of induction machines parameters estimation from measured slip curves, Arch. Elektrotech., **1**, 72, 1989.

[14] Kim D. H., Cho J. H.: Robust PID Controller Tuning Using Multiobjective Optimization Based on Clonal Selection of Immune Algorithm, 8th International Conference Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, Springer, 2004, pp. 50-56.

[15] Abraham A.: Intelligent Systems: Architectures and Perspectives, Recent Advances in Intelligent Paradigms and Applications, Abraham A., Jain L. and Kacprzyk J. (Eds.), Studies in Fuzziness and Soft Computing, Springer Verlag Germany, ISBN 3790815381, Chapter 1, 2002, pp. 1-35.

[16] Abraham A., Nath B.: A Neuro-Fuzzy Approach for Forecasting Electricity Demand In Victoria, Applied Soft Computing Journal, Elsevier Science, **1&2**, 2001, pp. 127-138.

[17] Abraham A.: Business Intelligence from Web Usage Mining, Journal of Information & Knowledge Management (JIKM), World Scientific Publishing Co., Singapore, **2**, 4, 2003, pp. 375-390.

[18] Abraham A., Philip N. S., Saratchandran P.: Modeling Chaotic Behavior of Stock Indices Using Intelligent Paradigms, International Journal of Neural, Parallel & Scientific Computations, USA, **11**, (1&2), 2003, pp. 143-160.